

## **REMARKS**

Applicants cancel dependent claim 8 and amend independent claim 7. No new matter is added by the amended claim, which is fully supported by the specification. Claims 1-7 and 9-26 remain pending.

### **Claim Rejections Under 35 U.S.C. § 103(a)**

The Examiner rejected claims 1-6 and 21-26 under 35 U.S.C. § 103(a) as being unpatentable over Gilbert et al. (U.S. Patent No. 5,121,498) in view of Gosling (U.S. Patent No. 5,668,999). Further, the Examiner rejected claims 7-20 under 35 U.S.C. § 103(a) as being unpatentable over Gilbert et al. in view of Gosling and Cartwright, Jr. (U.S. Patent No. 6,075,942). Applicants respectfully traverse.

Specifically, Gosling does not teach or suggest features and functional characteristics recited in independent claims 1, 7, 15, and 21. The reference teaches pre-verification of stack usage in bytecode program loops. To perform the pre-verification, the reference teaches various structures such as a virtual stack, an operand stack, a virtual local variable array, and a stack snapshot storage structure. Using these structures, Gosling discloses “the virtual stack stores a data type indicator value ...to be stored in the operand stack.” Column 5, lines 35-40. In order to access the values in the stacks, the reference teaches popping and pushing datum off and on the stacks, respectively, to perform verification. As further taught by the reference, the interpreter “must continually monitor the operand stack for overflows and underflows.” Column 4, lines 60-64. Thus, as taught and suggested by Gosling, because of possible stack overflows and underflows, there must be tests to monitor for *stack overflows* and *stack underflows*. Figure 4C, 452 and Figure 4D, 472. AS understood by those of ordinary skill in the art, stack overflows and stack underflows

cause disruptions in normal computer operations. Thus, stack monitoring prevents abnormal computer operations.

However, what the reference does not teach or suggest is “testing indexing expressions for underflow.” When Gosling discloses that the verifier of the “preferred embodiment of the present invention only seeks to verify *stack* manipulations” as illustrated by Figures 4A through 4G and column 7, lines 5-10, the reference discloses that the purpose of testing for overflows and underflows is to monitor *stack* overflows and underflows. In contrast, the independent claims recite creating a pre-loop structure capable of testing *indexing expressions* for underflow, generating a main loop structure, and creating a post-loop structure. Thus, multiple loop structures are created from an original loop structure. Multiple loop structures such as the pre-loop structure and the post-loop structure are capable of testing indexing expressions for overflow and underflow.

To the contrary, the reference tests stacks for underflow and overflow for monitoring purposes to prevent abnormal computer operations. Thus, the functional purpose of monitoring stacks during stack operations to access data stored in the stack does not provide motivation to test indexing expressions of multiple loop structures for overflow or underflow.

Regarding Cartwright, Jr., the reference does not disclose the features of “creating a range check elimination loop structure.” Specifically, the reference at column 7, lines 25-40 and column 9, lines 50-55, does not teach or suggest creating a range check elimination loop structure including a pre-loop structure, a main loop structure, and a post-loop structure. Instead, Cartwright, Jr. teaches the “*rearrangement* of computations.” Column 7, lines 29-30. Further, the goal of Cartwright, Jr. “concerns arrangements for compiling in one machine, programs to be

executed by another.” Column 1, lines 8-9. These rearrangements do not create a range check elimination loop structure because the computations are rearranged to avoid time-consuming computations. The rearrangement does not create multiple loop structures such as the pre-loop structure and the post-loop structure. Thus, because the reference does not teach or suggest creating a range check elimination loop structure, but only teaches the rearrangement of computations, Cartwright, Jr. does not render the Applicants’ claimed invention obvious.

Accordingly, because Gosling and Cartwright, Jr., singly or in combination with each other or Gilbert et al., do not teach or suggest the features recited by independent claims 1, 7, 15, and 21, Applicants respectfully submit that the independent claims are allowable. Further because claims 2-6, 9-14, 16-20, and 22-26 depend from the allowable independent claims, the dependent claims are allowable for at least the same reasons. Thus, Applicants respectfully request withdrawal of the 35 U.S.C. § 103(a) rejection.

Application No. 09/872,458.  
Non-Final Office Action mailed 2/25/2004.  
Response to the Non-Final Office Action mailed 5/25/04.

Applicants respectfully request a Notice of Allowance based on the foregoing remarks. If the Examiner has any questions concerning the present amendment, the Examiner is kindly requested to contact the undersigned at (408) 749-6900. If any other fees are due in connection with filing this amendment, the Commissioner is also authorized to charge Deposit Account No. 50-0805 (Order No. SUNMP018). A copy of the transmittal is enclosed for this purpose.

Respectfully submitted,  
MARTINE & PENILLA, LLP

A handwritten signature in black ink, appearing to read 'Feb Cabrasawan', followed by a horizontal line.

Feb Cabrasawan  
Reg. No. 51,521

Martine & Penilla, LLP  
710 Lakeway Drive, Suite 170  
Sunnyvale, California 94086  
Tel: (408) 749-6900  
**Customer Number 32291**